BEST AVAILABLE IMAGES

Defective images within this document are accurate representation of The original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

PATENT COOPERATION TREATY

From the INTERNATIONAL BUREAU

PCT

NOTIFICATION OF ELECTION

(PCT Rule 61.2)

To:

Commissioner
US Department of Commerce
United States Patent and Trademark Office, PCT
2011 South Clark Place Room CP2/5C24
Arlington, VA 22202
ETATS-UNIS D'AMERIQUE

in its capacity as elected Office

Date of mailing (day/month/year) 29 May 2001 (29.05.01)	
International application No. PCT/IL00/00540	**Applicant's or agent's file reference** 39279
International filing date (day/month/year) 07 September 2000 (07.09.00)	**Priority date** (day/month/year) 08 September 1999 (08.09.99)
Applicant KAGAN, Michael et al	

1. The designated Office is hereby notified of its election made:

[X] in the demand filed with the International Preliminary Examining Authority on:

26 March 2001 (26.03.01)

[ ] in a notice effecting later election filed with the International Bureau on:

2. The election [X] was

[ ] was not

made before the expiration of 19 months from the priority date or, where Rule 32 applies, within the time limit under Rule 32.2(b).

The International Bureau of WIPO **34, chemin des Colombettes** **1211 Geneva 20, Switzerland**	Authorized officer Claudio Borton
Facsimile No.: (41-22) 740.14.35	Telephone No.: (41-22) 338.83.38

PATENT COOPERATION TREATY

From the INTERNATIONAL BUREAU

PCT

NOTIFICATION OF RECEIPT OF RECORD COPY

(PCT Rule 24.2(a))

To:

COLB, Sanford, T.
Sanford T. Colb & CO.
P.O. Box 2273
76122 Rehovot
ISRAËL

Date of mailing (day/month/year)	IMPORTANT NOTIFICATION
23 October 2000 (23.10.00)	
Applicant's or agent's file reference	**International application No.**
39279	PCT/IL00/00540

The applicant is hereby notified that the International Bureau has received the record copy of the international application as detailed below.

Name(s) of the applicant(s) and State(s) for which they are applicants:

MELLANOX TECHNOLOGIES LTD. (for all designated States except US)
KAGAN, Michael et al (for US)

International filing date	:	07 September 2000 (07.09.00)
Priority date(s) claimed	:	08 September 1999 (08.09.99) 10 January 2000 (10.01.00) 27 April 2000 (27.04.00)
Date of receipt of the record copy by the International Bureau	:	25 September 2000 (25.09.00)
List of designated Offices	:	

AP :GH,GM,KE,LS,MW,MZ,SD,SL,SZ,TZ,UG,ZW
EA :AM,AZ,BY,KG,KZ,MD,RU,TJ,TM
EP :AT,BE,CH,CY,DE,DK,ES,FI,FR,GB,GR,IE,IT,LU,MC,NL,PT,SE
OA :BF,BJ,CF,CG,CI,CM,GA,GN,GW,ML,MR,NE,SN,TD,TG
National :AE,AG,AL,AM,AT,AU,AZ,BA,BB,BG,BR,BY,BZ,CA,CH,CN,CR,CU,CZ,DE,DK,DM,DZ,EE,ES,FI,GB,GD,GE,GH,GM,HR,HU,ID,IL,IN,IS,JP,KE,KG,KP,KR,KZ,LC,LK,LR,LS,LT,LU,LV,MA,MD,MG,MK,MN,MW,MX,MZ,NO,NZ,PL,PT,RO,RU,SD,SE,SG,SI,SK,SL,TJ,TM,TR,TT,TZ,UA,UG,US,UZ,VN,YU,ZA,ZW

The International Bureau of WIPO
34, chemin des Colombettes
1211 Geneva 20, Switzerland

Facsimile No. (41-22) 740.14.35

Authorized officer:

Céline Faust

Telephone No. (41-22) 338.83.38

Continuation of Form PCT/IB/301

N[illegible]ICATION OF RECEIPT OF RECORD C[illegible]

Date of mailing (day/month/year)	IMPORTANT NOTIFICATION
23 October 2000 (23.10.00)	
Applicant's or agent's file reference	**International application No.**
39279	PCT/IL00/00540

ATTENTION

The applicant should carefully check the data appearing in this Notification. In case of any discrepancy between these data and the indications in the international application, the applicant should immediately inform the International Bureau.

In addition, the applicant's attention is drawn to the information contained in the Annex, relating to:

- [X] time limits for entry into the national phase
- [ ] confirmation of precautionary designations
- [X] requirements regarding priority documents

A copy of this Notification is being sent to the receiving Office and to the International Searching Authority.

PATENT COOPERATION TREATY

PCT

REC'D 01 NOV 2001
WIPO PCT

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Article 36 and Rule 70)

14

Applicant's or agent's file reference 6727/2J027WO	**FOR FURTHER ACTION**	See Notification of Transmittal of International Preliminary Examination Report (Form PCT/IPEA/416)
International application No. PCT/IL00/00540	International filing date *(day/month/year)* 07 September 2000 (07.09.2000)	Priority date *(day/month/year)* 08 September 1999 (08.09.1999)
International Patent Classification (IPC) or national classification and IPC IPC(7): G06F 13/00 and US Cl.: 710/260		
Applicant MELLANOX TECHNOLOGIES LTD.		

1. This international preliminary examination report has been prepared by this International Preliminary Examining Authority and is transmitted to the applicant according to Article 36.

2. This REPORT consists of a total of [6] 5 sheets, including this cover sheet.

   ☐ This report is also accompanied by ANNEXES, i.e., sheets of the description, claims and/or drawings which have been amended and are the basis for this report and/or sheets containing rectifications made before this Authority (see Rule 70.16 and Section 607 of the Administrative Instructions under the PCT).

   These annexes consist of a total of ∅ sheets.

3. This report contains indications relating to the following items:

   - I ☒ Basis of the report
   - II ☐ Priority
   - III ☐ Non-establishment of report with regard to novelty, inventive step and industrial applicability
   - IV ☐ Lack of unity of invention
   - V ☒ Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
   - VI ☐ Certain documents cited
   - VII ☐ Certain defects in the international application
   - VIII ☐ Certain observations on the international application

Date of submission of the demand	Date of completion of this report
26 March 2001 (26.03.2001)	26 September 2001 (26.09.2001)
Name and mailing address of the IPEA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer Glenn A. Auve James R. Matthews Telephone No. (703) 305-3900

Form PCT/IPEA/409 (cover sheet)(July 1998)

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/IL00/00540

I. Basis of the report

1. With regard to the **elements** of the international application:*

☒ the international application as originally filed.

☒ the description:
pages 1-11 as originally filed
pages NONE , filed with the demand
pages NONE , filed with the letter of _____.

☒ the claims:
pages 12-15 , as originally filed
pages NONE , as amended (together with any statement) under Article 19
pages NONE , filed with the demand
pages NONE , filed with the letter of _____.

☒ the drawings:
pages 1-2 , as originally filed
pages NONE , filed with the demand
pages NONE , filed with the letter of _____.

☐ the sequence listing part of the description:
pages NONE , as originally filed
pages NONE , filed with the demand
pages NONE , filed with the letter of _____.

2. With regard to the **language**, all the elements marked above were available or furnished to this Authority in the language in which the international application was filed, unless otherwise indicated under this item.
These elements were available or furnished to this Authority in the following language _____ which is:

☐ the language of a translation furnished for the purposes of international search (under Rule23.1(b)).

☐ the language of publication of the international application (under Rule 48.3(b)).

☐ the language of the translation furnished for the purposes of international preliminary examination(under Rules 55.2 and/or 55.3).

3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international preliminary examination was carried out on the basis of the sequence listing:

☐ contained in the international application in printed form.

☐ filed together with the international application in computer readable form.

☐ furnished subsequently to this Authority in written form.

☐ furnished subsequently to this Authority in computer readable form.

☐ The statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.

☐ The statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished.

4. ☐ The amendments have resulted in the cancellation of:

☐ the description, pages NONE

☐ the claims, Nos. NONE

☐ the drawings, sheets/~~fig~~ NONE

5. ☒ This report has been established as if (some of) the amendments had not been made, since they have been considered to go beyond the disclosure as filed, as indicated in the Supplemental Box (Rule 70.2(c)).**

** Replacement sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to this report since they do not contain amendments (Rules 70.16 and 70.17).*
*** Any replacement sheet containing such amendments must be referred to under item 1 and annexed to this report.*

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/IL00/00540

V. Reasoned statement under Rule 66.2(a)(ii) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1. STATEMENT

Novelty (N)	Claims	4-6,9,10,14-16,18,21, and 27	YES
	Claims	1-3,7,8,11-13,17,19,20, and 22-26	NO
Inventive Step (IS)	Claims	4-6,9,10,14-16,18,21, and 27	YES
	Claims	1-3,7,8,11-13,17,19,20, and 22-26	NO
Industrial Applicability (IA)	Claims	1-27	YES
	Claims	NONE	NO

2. CITATIONS AND EXPLANATIONS
Please See Continuation Sheet

Supplemental Box
(To be used when the space in any of the preceding boxes is not sufficient)

Claims 4-6,9,10,14-16,18,21, and 27 meet the criteria set out in PCT Article 33(2)-(4), because the prior art does not teach or fairly suggest the following limitations.

With regard to claims 4,5,14, and 15, the prior art does not teach or fairly suggest reading the cause of the interrupt and incorporating the cause in the interrupt packet.

With regard to claims 6 and 16, the prior art does not teach or fairly suggest sending the interrupt packet after receiving an acknowledgment from the memory that the data have been written thereto.

With regard to claims 9 and 10, the prior art does not teach or fairly suggest passing the data to a system controller, wherein notifying the CPU comprises informing the CPU when an acknowledgment is received by the host network interface from the system controller.

With regard to claim 18, the prior art does not teach or fairly suggest a switch coupling the target channel adapter and the processor to the network, wherein the switch comprises a receive queue into which the target channel adapter places the data packets and the processor places the interrupt packet into the receive queue following the data packets.

With regard to claims 21 and 27, the prior art does not teach or fairly suggest that the channel adapter is an InfiniBand adapter.

Claims 1-3,7,8,11-13,17,19,20, and 22-26 lack novelty under PCT Article 33(2) as being anticipated by Gentry et al., U.S. Pat. No. 5,659,758.

As per claim 1, Gentry et al. (Gentry) shows receiving data from a peripheral device for transmission via a network to a memory associated with a CPU; receiving an interrupt signal from the peripheral associated with the data; sending one or more data packets containing the data over the network to a host interface serving the memory and CPU; and sending an interrupt packet over the network to the host interface, responsive to which an interrupt input of the CPU is asserted only after the one or more data packets have arrived at the host (figs. 3 and 6, abstract, and cols. 2 and 4-5). Gentry shows all of the steps recited in claim 1.

As for claim 2, the argument for claim 1 applies. Gentry also shows that the receiving of the data comprises receiving parallel data over a local bus from the peripheral (cols. 4-5). Gentry shows all of the steps recited in claim 2.

As for claim 3, the argument for claim 1 applies. Gentry also shows that receiving the data comprises receiving data to be written to the memory by direct memory access (col.2, lines 54-59). Gentry shows all of the steps recited in claim 3.

As for claim 7, the argument for claim 1 applies. Gentry also shows sending the data packets comprises sending data packets over a

Supplemental Box
(To be used when the space in any of the preceding boxes is not sufficient)

selected channel and sending the interrupt packet over the selected channel following the data packets (cols. 2 and 4-5). Gentry shows all of the steps recited in claim 7.

As for claim 8, the argument for claim 1 applies. Gentry also shows receiving the data packets and the interrupt packet at the host network interface; conveying the data packets for delivery to the memory over a local bus coupling the host interface with the memory and CPU; and notifying the CPU when all of the data have been conveyed (cols. 2 and 4-5). Gentry shows all of the steps recited in claim 8.

As for claim 11, the argument for claim 8 applies. Gentry also shows that notifying the CPU comprises asserting the interrupt input of the CPU responsive to receiving the interrupt packet at the host interface (cols. 2 and 4-5). Gentry shows all of the steps recited in claim 11.

As per claim 12, Gentry shows a target channel adapter operative to receive data from a peripheral device for transmission via a packet switching network to a memory associated with a central processing unit and to send one or more data packets over the network to a host interface; a target interface processor for receiving an interrupt signal from the peripheral and to send an interrupt packet over the network to the host interface responsive to which an interrupt input of the CPU is asserted only after the data packets have arrived at the host interface (abstract and cols. 2 and 4-5). Gentry shows all of the elements recited in claim 12.

As for claim 13, the argument for claim 12 applies. Gentry also shows that the target channel adapter comprises an interface to a local parallel bus linked to the peripheral device (cols. 4-5). Gentry shows all of the elements recited in claim 13.

As for claim 17, the argument for claim 12 applies. Gentry also shows that the target channel adapter is coupled to send the data packets over a selected channel through the network and the processor is adapted to send the interrupt packet over the selected channel following the data packets (cols. 2 and 4-5). Gentry shows all of the elements recited in claim 17.

As for claim 19, the argument for claim 12 applies. Gentry also shows a host interface which is coupled to receive the data and interrupt packets and is operative to convey the data for delivery to the memory over a local bus and to notify the CPU when all the data have been conveyed (cols. 2 and 4-5). Gentry shows all of the elements recited in claim 19.

As for claim 20, the argument for claim 19 applies. Gentry also shows that the host interface is coupled to assert the interrupt to the CPU responsive to the interrupt packet (cols. 2 and 4-5). Gentry shows all of the elements recited in claim 20.

As per claim 22, Gentry shows a host channel adapter for receiving data packets transmitted over a network from a peripheral, and to convey data from the packets for delivery to a memory associated with a CPU over a local bus that is coupled to the memory and the CPU, and further to receive an interrupt packet over the network responsive to an interrupt signal asserted by the peripheral device after sending the data to the network; and a host interface processor responsive to the interrupt packet to notify the CPU when all of the data have been conveyed (abstract and cols. 2 and 4-5). Gentry shows all of the elements recited in claim 22.

As for claim 23, the argument for claim 22 applies. Gentry also shows that the host channel adapter is operative to convey the data by direct memory access (col.2). Gentry shows all of the elements recited in claim 23.

As for claim 24, the argument for claim 22 applies. Gentry also shows that the host channel adapter is operative to convey the data to a system controller on the bus and wherein the CPU is notified when an acknowledgment is received by the host channel adapter from the system controller (cols. 4-5). Gentry shows all of the elements recited in claim 24.

As for claim 25, the argument for claim 24 applies. Gentry also shows that the host interface processor is coupled to assert the interrupt input of the CPU after the acknowledgment has been received (cols. 4-5). Gentry shows all of the elements recited in claim 25.

As for claim 26, the argument for claim 22 applies. Gentry also shows that the host interface processor is coupled to assert the interrupt input of the CPU responsive to receipt of the interrupt packet at the host interface (cols. 4-5). Gentry shows all of the elements recited in claim 26.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 March 2001 (15.03.2001)

PCT

(10) International Publication Number
WO 01/18654 A1

(51) International Patent Classification[7]: G06F 13/00

(21) International Application Number: PCT/IL00/00540

(22) International Filing Date:
7 September 2000 (07.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/152,849	8 September 1999 (08.09.1999)	US
60/175,339	10 January 2000 (10.01.2000)	US
09/559,352	27 April 2000 (27.04.2000)	US

(71) Applicant *(for all designated States except US)*: **MELLANOX TECHNOLOGIES LTD.** [IL/IL]; P.O. Box 86, 20692 Yokneam (IL).

(72) Inventors; and

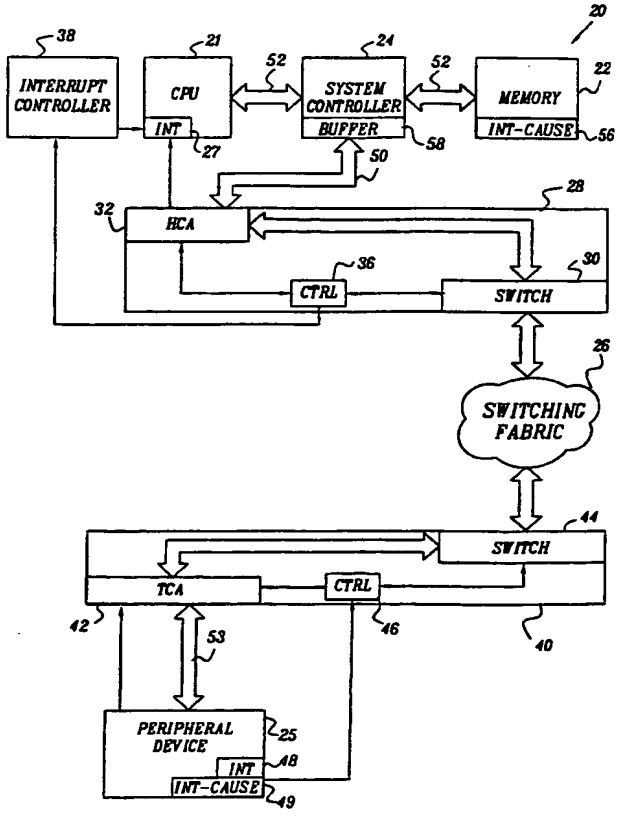
(75) Inventors/Applicants *(for US only)*: **KAGAN, Michael** [IL/IL]; Hashomer Street 71, 30900 Zichron Yaakov (IL). **CRUPNICOFF, Diego** [AR/IL]; Sitvanit Street 9, 34793 Haifa (IL). **GABBAY, Freddy** [IL/IL]; Derech Hashalom Street 75/2, 67942 Tel Aviv (IL). **ROTTENBERG, Shimon** [IL/IL]; Dubnov Street 27, 64597 Tel Aviv (IL).

(74) Agents: **COLB, Sanford, T.** et al.; Sanford T. Colb & CO., P.O. Box 2273, 76122 Rehovot (IL).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

[Continued on next page]

(54) Title: SYNCHRONIZATION OF INTERRUPTS WITH DATA PACKETS



(57) Abstract: A method and apparatus for conveying data over a packet-switching network (26). Data are received from a peripheral device (25) for transmission via the network to a memory (22) associated with a central processing unit (CPU) (21), followed by an interrupt signal from the peripheral device associated with the data. One or more data packets containing the data are sent over the network to a host network interface (32) serving the memory and the CPU, followed by an interrupt packet sent over the network to the host network interface. Responsive to the interrupt packet, an interrupt input of the CPU is asserted only after the one or more data packets have arrived at the host network interface.

WO 01/18654 A1

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYNCHRONIZATION OF INTERRUPTS WITH DATA PACKETS

CROSS-REFERENCE TO RELATED APPLICATION

This application claims the benefit of U.S. Provisional Patent Application 60/152,849, filed September 8, 1999, and of U.S. Provisional Patent Application 60/175,339, filed January 10, 2000. Both of these co-pending applications are assigned to the assignee of the present patent application and are incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates generally to computing systems, and specifically to systems that use packet-switching fabrics to connect a computer host to peripheral devices.

BACKGROUND OF THE INVENTION

In current-generation computers, the central processing unit (CPU) is connected to the system memory and to peripheral devices by a parallel bus, such as the ubiquitous Peripheral Component Interface (PCI) bus. As data path-widths grow, and clock speeds become faster, however, the parallel bus is becoming too costly and complex to keep up with system demands. In response, the computer industry is moving toward fast, packetized, serial input/output (I/O) bus architectures, in which computing hosts and peripheral are linked by a switching network, commonly referred to as a switching fabric. A number of architectures of this type have been proposed, including "Next Generation I/O" (NGIO) and "Future I/O" (FIO), culminating in the "InfiniBand" architecture, which has been advanced by a consortium led by a group of industry leaders (including Intel, Sun, Hewlett Packard, IBM, Compaq, Dell and Microsoft). Storage Area Networks (SAN) provide a similar, packetized, serial approach to high-speed storage access, which can also be implemented using an InfiniBand fabric.

In a parallel bus-based computer system, when a peripheral device needs to deliver data to the CPU, it typically writes the data to the memory over the bus, using direct memory access. When the peripheral has finished writing, it asserts an interrupt to the CPU on one of the interrupt lines of the bus. Bus arbitration ensures that the CPU will not attempt to read the data from the memory until the writing of the data is complete. On the other hand, when the peripheral device and the CPU are connected by a packet-switching fabric, such as an InfiniBand fabric, they operate asynchronously. Furthermore, the data sent to the memory and the interrupt to the CPU travel over different paths, or channels. Typically, a separate line or

channel is provided to connect the interrupt pin of the peripheral device to an interrupt controller of the CPU, bypassing the switching fabric. Therefore, there is no *a priori* assurance that all of the data will have been written to the memory before the CPU begins reading.

The "race" between the interrupt path and the data path can result in errors (as when a CPU read stalls the data). Care must therefore be taken to synchronize data and interrupt handling and to make sure that the data have been completely written to the memory before the CPU attempts to read it.

A common solution in this situation is to program the CPU to access the peripheral device before accessing the memory, typically by performing a "configuration read" from the peripheral device. In this mode of operation, after the peripheral device has asserted the interrupt to the CPU (indicating that the last item of data has been sent to the memory), the CPU issues a read request through the switching fabric, to read an interrupt cause register in the peripheral device. The peripheral device responds to the read request by sending a packet containing the interrupt cause to the CPU over the same channel as it used to send the data to the memory. Since packets are ordered within a channel, the response to configuration read arrives at the CPU after all of the previous writes have been flushed to memory. The CPU begins to read the data from the memory only after it has received the interrupt cause packet back from the peripheral device. The configuration read thus serves two crucial purposes: it provides the CPU with the cause information that it needs in order to serve the interrupt, and it ensures that the CPU reads the memory only after all of the data have been written there.

This scheme has a number of serious performance drawbacks, however. Every interrupt sent by the peripheral device necessitates an additional exchange of messages through the switching fabric between the CPU and peripheral device. The exchange adds substantial latency – typically 10 microseconds or more - every time the CPU must service an interrupt. Furthermore, since configuration reads are used as synchronization barriers, the CPU is stalled from the moment the configuration read request is issued until its response has arrived. Valuable CPU time is therefore wasted waiting for the interrupt cause to be retrieved.

U.S. Patent 5,689,713, whose disclosure is incorporated herein by reference, describes a method for interrupt request handling in a packet-switched computer system. The system may include a number of interrupt sources, which direct interrupts to any of a number of interrupt handlers. A system controller acts as an intermediary between interrupting devices and

"interruptees." It includes an interrupt queue coupled to each interrupt source for receiving multiple interrupt requests, and an output queue coupled to each interrupt handler. The controller thus enables asynchronous data from multiple sources to be conveyed across a packet-switched interconnection, while providing a dedicated channel for interrupts associated with the data packets.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide an improved method and system for passing data packets and associated interrupts through a switching fabric.

It is a further object of some aspects of the present invention to provide a method and system for communication between a CPU and peripheral devices via a switching fabric that ensures proper synchronization between data and interrupts transmitted over the fabric.

It is still a further object of some aspects of the present invention to provide a method and system for communication between a CPU and peripheral devices via a switching fabric that reduces latency and processing time required for servicing of interrupts by the CPU.

In preferred embodiments of the present invention, a CPU and a peripheral device are linked to a packet- switching fabric by respective host and target network interfaces. The target interface receives data over a local bus from the peripheral device, for transmission in the form of packets to a system memory associated with the CPU. After sending the data, the peripheral device asserts an interrupt. The interrupt from the device is connected to an interrupt input of the target interface, rather than directly to the CPU or to a central system controller, as in systems known in the art. In response to the interrupt, the target interface reads the interrupt cause from the peripheral device, and then sends a special interrupt packet, including the interrupt cause, to the host interface. Preferably, the target interface sends the interrupt packet on the same channel as it sent the data packets, i.e., over the same "virtual lane," or route, and with the same priority as the data packets. It thus assures that the host interface will receive the interrupt packet only after it has received all of the preceding data packets.

Upon receiving the interrupt packet, the host interface places the interrupt cause in a predefined register in the memory. An interrupt signal is then sent from the host interface to an interrupt input of the CPU. Upon receiving the signal, the CPU checks to ensure that the host interface has finished writing all of the data from the peripheral device to the memory. This

check serves a similar purpose to the configuration read described in the Background of the Invention. Only after completing the check does the CPU read the interrupt cause and begin processing the data in the memory. The CPU performs all of these steps locally, communicating with the host interface and memory over a local system bus, with latency on the order of nanoseconds, rather than having to exchange messages with the peripheral device through the switching fabric, taking many microseconds. As a result, interrupt response latency is minimized, and the CPU does not waste precious time and resources waiting for the configuration read response.

In preferred embodiments of the present invention, the switching fabric comprises an InfiniBand network, and the host and target interfaces respectively comprise host and target channel adapters. It will be appreciated, however, that the principles of the present invention may similarly be applied to transmission of interrupts through substantially any packet-switched network.

There is therefore provided, in accordance with a preferred embodiment of the present invention, a method for conveying data over a packet-switching network, including:

receiving data from a peripheral device for transmission via the network to a memory associated with a central processing unit (CPU);

receiving an interrupt signal from the peripheral device associated with the data;

sending one or more data packets containing the data over the network to a host network interface serving the memory and the CPU; and

sending an interrupt packet over the network to the host network interface, responsive to which an interrupt input of the CPU is asserted only after the one or more data packets have arrived at the host network interface.

Typically, receiving the data includes receiving parallel data over a local bus from the peripheral device. Additionally or alternatively, receiving the data includes receiving data to be written to the memory by direct memory access.

Preferably, sending the interrupt packet includes reading a cause of the interrupt from the peripheral device, and incorporating the cause in the interrupt packet. Further preferably, the method includes receiving the interrupt packet at the host network interface, and writing the cause to a predetermined address in the memory, to be read by the CPU after the interrupt input is asserted.

In a preferred embodiment, sending the interrupt packet includes sending the interrupt packet after receiving an acknowledgment from the memory that the data have been written thereto.

Preferably, sending the one or more data packets includes sending the data packets over a selected channel through the network, and sending the interrupt packet includes sending the interrupt packet over the selected channel following the data packets.

Further preferably, the method includes:

receiving the data packets and the interrupt packet at the host network interface;

conveying the data in the packets for delivery to the memory over a local bus coupling the host network interface to the memory and the CPU; and

notifying the CPU when all of the data have been conveyed.

Most preferably, conveying the data in the packets includes passing the data to a system controller on the bus, and notifying the CPU includes informing the CPU when an acknowledgment is received by the host network interface from the system controller, typically by asserting the interrupt input of the CPU after the acknowledgment from the system controller has been received. Additionally or alternatively, notifying the CPU includes asserting the interrupt input of the CPU responsive to receiving the interrupt packet at the host network interface.

There is also provided, in accordance with a preferred embodiment of the present invention, network interface apparatus, including:

a target channel adapter, which is operative to receive data from a peripheral device for transmission via a packet-switching network to a memory associated with a central processing unit (CPU) and to send one or more data packets containing the data over the network to a host network interface serving the memory and the CPU; and

a target interface processor, adapted to receive an interrupt signal from the peripheral device associated with the data, and to send an interrupt packet over the network to the host network interface, responsive to which an interrupt input of the CPU is asserted only after the one or more data packets have arrived at the host network interface.

There is further provided, in accordance with a preferred embodiment of the present invention, network interface apparatus, including:

a host channel adapter, which is operative to receive data packets transmitted over a packet-switching network from a peripheral device, and to convey data from the packets for delivery to a memory associated with a CPU over a local bus that is coupled to the memory and the CPU, and further to receive an interrupt packet sent over the network responsive to an interrupt signal asserted by the peripheral device after sending the data to the network; and

a host interface processor, adapted, responsive to the interrupt packet, to notify the CPU when all of the data have been conveyed to the local bus.

Preferably, the target and host channel adapters include InfiniBand adapters.

The present invention will be more fully understood from the following detailed description of the preferred embodiments thereof, taken together with the drawings in which:

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram that schematically illustrates a computing system based on a packet- switching fabric, in accordance with a preferred embodiment of the present invention;

Fig. 2 is a flow chart that schematically illustrates a method for transmitting data from a peripheral device to a CPU in the system of Fig. 1, in accordance with a preferred embodiment of the present invention; and

Fig. 3 is a flow chart that schematically illustrates a method for processing data received by the CPU in the system of Fig. 1, in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Fig. 1 is a block diagram that schematically illustrates a computing system 20 built around a switching fabric 26, in accordance with a preferred embodiment of the present invention. The switching fabric preferably comprises an InfiniBand fabric, as described in the Background of the Invention, and some of the terms used hereinbelow are specific to the InfiniBand architecture. It will be understood, however, that the system architecture and methods of communication described herein are in no way limited to InfiniBand, and that other switching fabrics, as are known in the art, may be configured to handle and convey interrupts in a similar manner.

A CPU 21 is coupled to communicate via a system bus 52 with a system controller 24 and a system memory 22, as is known in the art. Typically (although not necessarily), the CPU comprises an Intel Pentium processor, and bus 52 is a proprietary bus used in conjunction with

this processor. System controller 24 is coupled to a standard I/O bus 50, such as a PCI bus, for the purpose of communicating with peripheral devices, such as I/O adapters of various types. One such peripheral device 25 is shown in Fig. 1 by way of example, but in practical applications, system 20 typically comprises multiple peripheral devices and, possibly, multiple CPUs. Peripheral device 25 includes an interrupt output 48, which it asserts in order to gain the attention of the CPU. In systems known in the art, interrupt output 48 is connected directly to an interrupt controller 38, such as an Intel 8259 device, which actuates an appropriate interrupt input 27 of CPU 21 when the interrupt is asserted. In system 20, however, interrupt output 48 and input 27 are linked only through fabric 26, as described hereinbelow.

Bus 50 is coupled to fabric 26 by a host network interface unit 28. This unit comprises a host channel adapter (HCA) 32, which interfaces with bus 50 and converts data between packet and parallel forms. Alternatively, the HCA may be designed to interface with system bus 52. A switch 30 links the HCA to one or more core switches in the fabric. Ordinarily, data in packets received by switch 30 from fabric 26 are passed through HCA 32 to bus 50. An exception is made, however, for management packets, which are packets that carry a special header identifying themselves as such and including a local identifier (LID) address of either switch 30 or HCA 32. These packets contain control instructions for the switch or HCA. They are placed in a dedicated register of the switch or HCA, as appropriate, which then attempts to decode the instructions and carry them out. Typically, the processing capabilities of the switch and HCA are very limited, and they are assisted by a fabric service agent (FSA), as described below, in dealing with at least some of these management packets.

A host interface unit controller 36 acts as the FSA in interface unit 28. The controller preferably comprises a microprocessor with random access memory (RAM) for software code and data, communicates with HCA 32 and switch 30. Alternatively, the controller may comprise a hard-wired hardware element or digital signal processor. When HCA 32 or switch 30 receives a management packet that it cannot decode, it passes the packet to the controller. The controller decodes the packet, preferably based on suitable software stored in its code RAM. It then takes whatever action is called for by the packet, such as giving appropriate instructions to HCA 32 or switch 30. When the HCA receives an interrupt packet, as described below, the actions taken by controller 36 also include signaling interrupt controller 38 via an interrupt output of unit 28, so as to actuate interrupt input 27 of CPU 21.

Although for simplicity, only a single interrupt line from unit 28 to controller 38 is shown in Fig. 1, the unit preferably comprises multiple interrupt lines. These lines can be actuated selectively by controller 36 so as to send multiple, different interrupts to CPU 21 depending on the content of interrupt packets received by the HCA. Alternatively or additionally, the different interrupt lines may be used to signal other host devices that are linked to bus 50.

Peripheral device 25 is coupled to fabric 26 by a target network interface unit 40, similar in structure to unit 28. A target channel adapter (TCA) 42 in unit 40 interfaces via an I/O bus 53 with device 25. Typically, although not necessarily, bus 53 comprises a PCI bus, like bus 50. A switch 44 links the TCA to the switching fabric. A target unit controller 46, similar to controller 36, acts as FSA to TCA 42 and switch 44 and also has a suitable input to receive signals from interrupt output 48 of device 25.

Fig. 2 is a flow chart that schematically illustrates a method by which target interface unit 40 processes and transmits data from peripheral device 25 to HCA 32 over fabric 26, in accordance with a preferred embodiment of the present invention. At a data writing step 60, device 25 writes data via bus 53 to TCA 42, to be conveyed by direct memory access to memory 22. The peripheral device assigns a priority to the data to be transmitted and informs the TCA of this priority. At a data sending step 62, the TCA packetizes the data and sends it over fabric 26 to the address of HCA 32, with the priority assigned by the peripheral device. A packet header instructs the HCA to write the data to memory 22. Preferably, the TCA negotiates with switch 44 and fabric 26 to assign a fixed route for all of the packets through the fabric. Such a route, together with the priority of the packets, is referred to herein as a channel. InfiniBand specifies that packets travelling over the same channel are always kept in their original order.

When device 25 has finished posting to TCA 42 all of the data that it has to send, it asserts interrupt output 48, at an interrupt assertion step 64. At the same time, the peripheral device places the cause for the interrupt (in this case, to instruct CPU 21 to read the data from memory 22) in an interrupt cause register 49. In systems known in the art, when the CPU receives the interrupt, it must communicate with the peripheral device in order to read this register. In system 20, however, the interrupt signal is received by controller 46, which instructs TCA 42 to read the interrupt cause from register 49, at a cause reading step 66.

Based on the interrupt cause information read by the TCA, controller 46 constructs an interrupt packet containing the interrupt cause information, at an interrupt packet sending step 68. The interrupt packet is a management packet addressed to the LID of HCA 32. It is preferably sent by controller 46 over the same channel, or virtual lane, as the data packets, after the last of the data packets has been sent. The interrupt packet also identifies the data with which the interrupt is associated. As a result, when the interrupt packet arrives at its destination, controller 36 will be able to generate an interrupt to CPU 21 that is associated with the appropriate memory write, as described below. Controller 46 assures than interrupt packet is sent to the fabric after all of the data packets have already been accepted for sending. It thus ensures that HCA 32 will receive the interrupt packet only after it has received all of the data packets.

As an alternative, controller 46 may delay sending the interrupt packet until TCA 42 receives an acknowledgment from memory 22 that it has received all of the data. This approach introduces additional delay before CPU 21 can receive and act upon the interrupt, but it obviates the need to ensure that the interrupt packet is routed over the same channel as the data packets. Such an approach may be called for in particular when switching fabric 26 comprises a network in which consistent routing and ordering are not necessarily maintained among successive packets. This approach can also be used when the interrupt path and data path are not the same, and fork at an earlier stage than in Fig. 1. Such path incongruity may occur, for example, when the device writing data to the memory is different from the device asserting the interrupt to the CPU. Sometimes it is also desirable to send interrupts on different (high-priority) routes, because data routes can be congested, causing interrupt messages to get stuck behind data.

Fig. 3 is a flow chart that schematically illustrates a method by which data and accompanying interrupt packets are received and processed by host interface unit 28 and CPU 21, in accordance with a preferred embodiment of the present invention. At a packet reception step 70, HCA 32 receives the data and interrupt packets sent from target interface unit 40. The HCA posts the data in the data packets via bus 50 to a buffer 58 of system controller 24. The system controller proceeds to write the data from its buffer to the appropriate addresses in memory 22, as is known in the art. The HCA passes the interrupt packet to controller 36 for

decoding, at an interrupt processing step 72. The controller extracts the cause of the interrupt and posts this information, via HCA 32, to an interrupt cause register 56 in memory 22.

Before CPU 21 services the interrupt represented by the interrupt packet, it is necessary to ensure that all of the associated data have been written to memory 22, at a delivery completion step 74. In the case that controller 46 of target interface unit 40 is programmed to send the interrupt packet only after receiving the acknowledgment from memory 22, as described above, this problem is already solved. Otherwise, controller 36 preferably waits to assert the interrupt until system controller 24 has acknowledged to HCA 32 that it has received all of the data. In response to this acknowledgment, controller 36 sends an interrupt signal to interrupt controller 38, at an interrupt assertion step 76. The interrupt controller actuates interrupt input 27 of CPU 21, to inform the CPU that an interrupt has arrived from HCA 32. In response to the interrupt, the CPU preferably sends a dummy read command to the HCA, in order to ensure that buffer 58 is flushed to memory 22 before the CPU itself begins to process the data in the memory.

As a further alternative, as long as it is assured that the interrupt packet reached HCA 32 after the last of the data packets (which will be the case when all of the packets are sent over the same channel, as described above), controller 36 may send the interrupt signal to interrupt controller 38 immediately, without waiting for an acknowledgment from system controller 24. In this case, upon receiving the interrupt, CPU 21 preferably sends a "fence" command to HCA 32. This command instructs the HCA to mark the last packet currently in its receive queue, and to inform the CPU when this last packet has been written to system controller 24. At this point, the CPU can send its dummy read command and begin processing the data in the memory.

Once it is assured that all of the relevant data have reached their destination in memory 22, CPU 21 reads the cause of the current interrupt from register 56, at a cause reading step 78. Based on this information, the CPU processes the data that peripheral device 25 has placed in the memory, at a data processing step 80. Unlike methods of interrupt processing known in the art, all of the steps in the method of Fig. 3 are carried out locally, typically over busses 50 and 52, without the need for messages to traverse fabric 26.

It will be appreciated that the preferred embodiments described above are cited by way of example, and that the present invention is not limited to what has been particularly shown

and described hereinabove. Rather, the scope of the present invention includes both combinations and subcombinations of the various features described hereinabove, as well as variations and modifications thereof which would occur to persons skilled in the art upon reading the foregoing description and which are not disclosed in the prior art.

CLAIMS

1. A method for conveying data over a packet-switching network, comprising:

receiving data from a peripheral device for transmission via the network to a memory associated with a central processing unit (CPU);

receiving an interrupt signal from the peripheral device associated with the data;

sending one or more data packets containing the data over the network to a host network interface serving the memory and the CPU; and

sending an interrupt packet over the network to the host network interface, responsive to which an interrupt input of the CPU is asserted only after the one or more data packets have arrived at the host network interface.

2. A method according to claim 1, wherein receiving the data comprises receiving parallel data over a local bus from the peripheral device.

3. A method according to claim 1, wherein receiving the data comprises receiving data to be written to the memory by direct memory access.

4. A method according to claim 1, wherein sending the interrupt packet comprises reading a cause of the interrupt from the peripheral device, and incorporating the cause in the interrupt packet.

5. A method according to claim 4, and comprising receiving the interrupt packet at the host network interface, and writing the cause to a predetermined address in the memory, to be read by the CPU after the interrupt input is asserted.

6. A method according to any of the preceding claims, wherein sending the interrupt packet comprises sending the interrupt packet after receiving an acknowledgment from the memory that the data have been written thereto.

7. A method according to any of claims 1-5, wherein sending the one or more data packets comprises sending the data packets over a selected channel through the network, and wherein sending the interrupt packet comprises sending the interrupt packet over the selected channel following the data packets.

8. A method according to any of claims 1-5, and comprising:

receiving the data packets and the interrupt packet at the host network interface;

conveying the data in the packets for delivery to the memory over a local bus coupling the host network interface to the memory and the CPU; and

notifying the CPU when all of the data have been conveyed.

9. A method according to claim 8, wherein conveying the data in the packets comprises passing the data to a system controller on the bus, and wherein notifying the CPU comprises informing the CPU when an acknowledgment is received by the host network interface from the system controller.

10. A method according to claim 9, wherein informing the CPU comprises asserting the interrupt input of the CPU after the acknowledgment from the system controller has been received.

11. A method according to claim 8, wherein notifying the CPU comprises asserting the interrupt input of the CPU responsive to receiving the interrupt packet at the host network interface.

12. Network interface apparatus, comprising:

a target channel adapter, which is operative to receive data from a peripheral device for transmission via a packet-switching network to a memory associated with a central processing unit (CPU) and to send one or more data packets containing the data over the network to a host network interface serving the memory and the CPU; and

a target interface processor, adapted to receive an interrupt signal from the peripheral device associated with the data, and to send an interrupt packet over the network to the host network interface, responsive to which an interrupt input of the CPU is asserted only after the one or more data packets have arrived at the host network interface.

13. Apparatus according to claim 12, wherein the target channel adapter comprises an interface to a local parallel bus linked to the peripheral device, over which the device sends the data.

14. Apparatus according to claim 12, wherein the target channel adapter is operative to read a cause of the interrupt from the peripheral device, and wherein the processor is adapted to incorporate the cause in the interrupt packet.

15. Apparatus according to claim 14, and comprising a host channel adapter, coupled to receive the interrupt packet at the host network interface, and to write the cause to a predetermined address in the memory, to be read by the CPU after the interrupt input is asserted.

16. Apparatus according to any of claims 12-15, wherein the processor is adapted to send the interrupt packet after receiving an acknowledgment from the memory that the data have been written thereto.

17. Apparatus according to any of claims 12-15, wherein the target channel adapter is coupled to send the data packets over a selected channel through the network, and wherein the processor is adapted to send the interrupt packet over the selected channel following the data packets.

18. Apparatus according to claim 17, and comprising a switch coupling the target channel adapter and the processor to the network, wherein the switch comprises a receive queue into which the target channel adapter places the data packets, and wherein the processor is adapted to place the interrupt packet into the receive queue following the data packets.

19. Apparatus according to any of claims 12-15, and comprising a host interface unit, which is coupled to receive the data and interrupt packets transmitted over the network, and is operative to convey the data in the packets for delivery to the memory over a local bus coupled to the memory and the CPU and to notify the CPU when all of the data have been conveyed.

20. Apparatus according to claim 19, wherein the host interface unit is coupled to assert the interrupt to the CPU responsive to the interrupt packet.

21. Apparatus according to any of claims 12-15, wherein the target channel adapter comprises an InfiniBand adapter.

22. Network interface apparatus, comprising:

a host channel adapter, which is operative to receive data packets transmitted over a packet-switching network from a peripheral device, and to convey data from the packets for delivery to a memory associated with a CPU over a local bus that is coupled to the memory and the CPU, and further to receive an interrupt packet sent over the network responsive to an interrupt signal asserted by the peripheral device after sending the data to the network; and

a host interface processor, adapted, responsive to the interrupt packet, to notify the CPU when all of the data have been conveyed to the local bus.

23. Apparatus according to claim 22, wherein the host channel adapter is operative to convey the data to the memory by direct memory access.

24. Apparatus according to claim 22, wherein the host channel adapter is operative to convey the data to a system controller on the bus, and wherein the CPU is notified when an acknowledgment is received by the host channel adapter from the system controller.

25. Apparatus according to claim 24, wherein the host interface processor is coupled to assert the interrupt input of the CPU after the acknowledgment from the system controller has been received.

26. Apparatus according to any of claims 22-25, wherein the host interface processor is coupled to assert the interrupt input of the CPU responsive to receipt of the interrupt packet at the host network interface.

27. Apparatus according to any of claims 22-25, wherein the host channel adapter comprises an InfiniBand adapter.

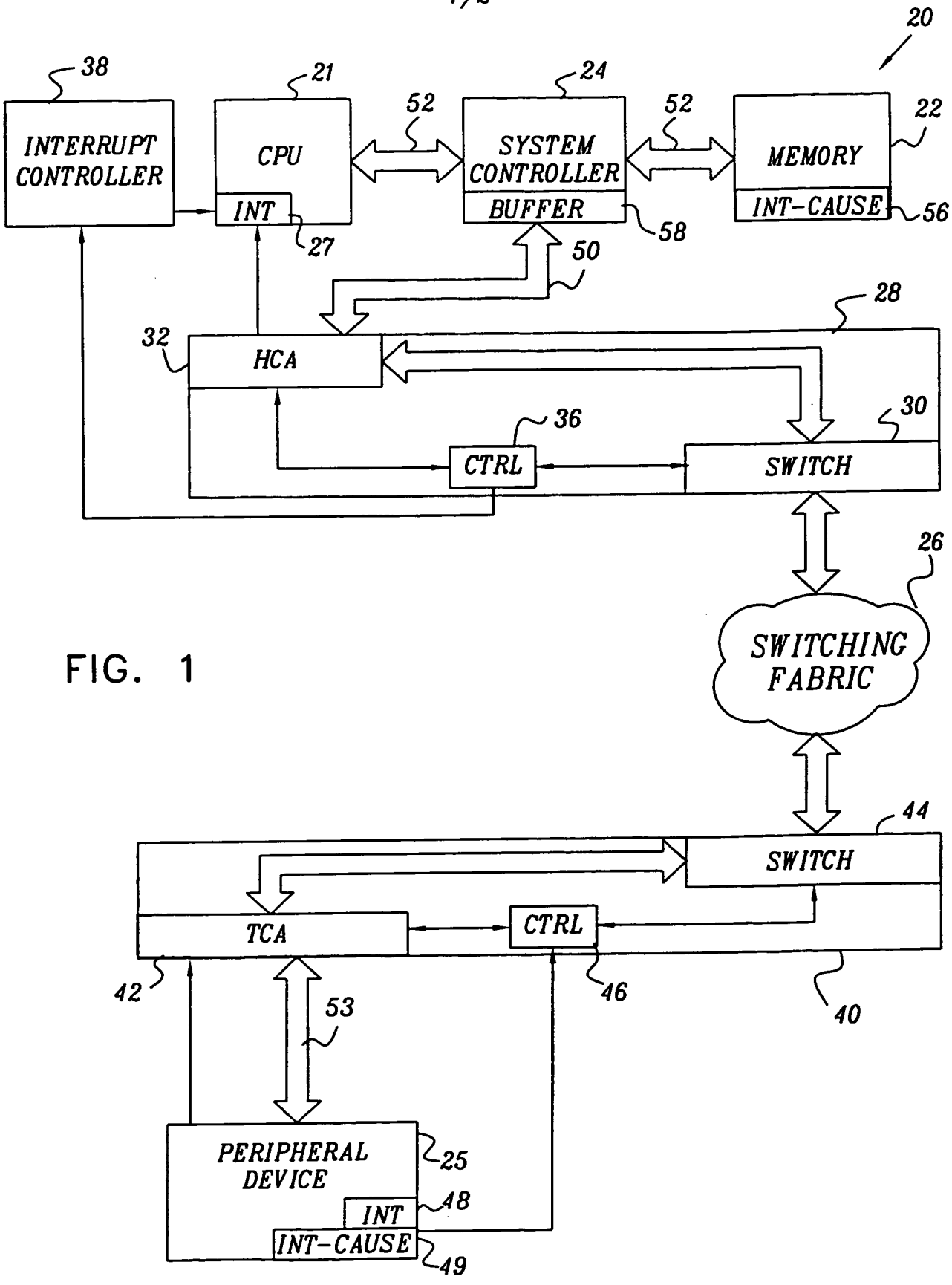

20
38
INTERRUPT CONTROLLER
21
CPU
INT
27
52
24
SYSTEM CONTROLLER
BUFFER
58
52
22
MEMORY
INT-CAUSE
56
50
28
32
HCA
36
CTRL
30
SWITCH
26
SWITCHING FABRIC
44
SWITCH
TCA
CTRL
42
46
40
53
PERIPHERAL DEVICE
25
INT
48
INT-CAUSE
49

FIG. 1

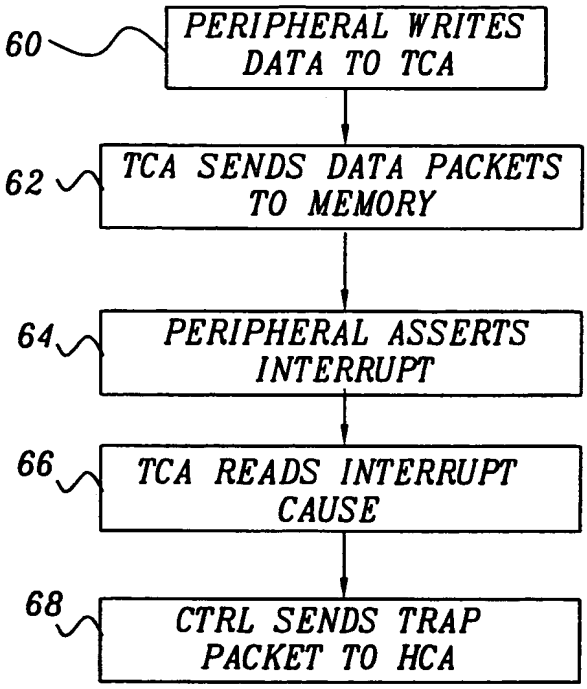

60
PERIPHERAL WRITES DATA TO TCA
62
TCA SENDS DATA PACKETS TO MEMORY
64
PERIPHERAL ASSERTS INTERRUPT
66
TCA READS INTERRUPT CAUSE
68
CTRL SENDS TRAP PACKET TO HCA

FIG. 2

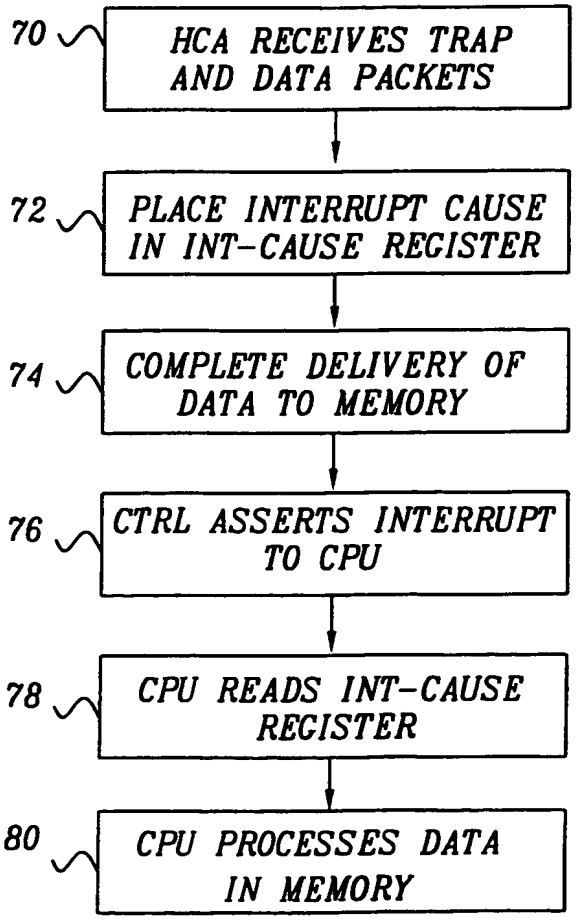

70
HCA RECEIVES TRAP AND DATA PACKETS
72
PLACE INTERRUPT CAUSE IN INT-CAUSE REGISTER
74
COMPLETE DELIVERY OF DATA TO MEMORY
76
CTRL ASSERTS INTERRUPT TO CPU
78
CPU READS INT-CAUSE REGISTER
80
CPU PROCESSES DATA IN MEMORY

FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IL00/00540

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 13/00
US CL : 710/260

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 710/260,263,266; 370/426; 709/238,250

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Please See Continuation Sheet

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,659,758 A (GENTRY et al) 19 August 1997, abstract, figs. 3 and 6, and columns 2, 4-5.	1-3,7,8,11-13,17,19,20, 22-26
A,P	US 6,038,629 A (OGILVIE et al) 14 March 2000, abstract.	1-27
A	US 5,440,545 A (BUCHHOLZ et al) 08 August 1995, abstract and columns 1-2.	1-27
A	US 5,689,713 A (NORMOYLE et al) 18 November 1997, abstract.	1-27

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
15 November 2000 (15.11.2000)	28 DEC 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer Glenn A. Auve James R. Matthews Telephone No. (703) 305-3900

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IL00/00540

Continuation of B. FIELDS SEARCHED Item 3: EAST text search: channel adj adapter; packet adj switch$3 adj network; interrupt adj2 packet$1; infiniband
WWW search using Google search engine: infiniband